Using Class Frequency for Improving Centroid-based Text Classification

Verayuth Lertnattee[1] and Chanisara Leuviphan[1]
[1]Department of Health-related Informatics, Faculty of Pharmacy
Silpakorn University, Maung, Nakorn Pathom, 73000 Thailand
E-mail: verayuths@hotmail.com, verayuth@su.ac.th, chanisara@su.ac.th

***Abstract*— Most previous works on text classification, represented importance of terms by term occurrence frequency (tf) and inverse document frequency (idf). This paper presents the ways to apply class frequency in centroid-based text categorization. Three approaches are taken into account. The first one is to explore the effectiveness of inverse class frequency on the popular term weighting, i.e., TFIDF, as a replacement of idf and an addition to TFIDF. The second approach is to evaluate some functions, which are used to adjust the power of inverse class frequency. The other approach is to apply terms, which are found in only one class or few classes, to improve classification performance, using two-step classification. From the results, class frequency expresses its usefulness on text classification, especially the two-step classification.**

***Index Terms*—text classification, term weighting, class frequency, linear classifier**

I. Introduction

The increasing availability of online text information, there has been extreme need to find and organize relevant information in text documents. The automated text categorization (also known as text classification) becomes a significant tool to organize text documents efficiently. A variety of classification methods were developed and used in different schemes, such as probabilistic models [1], neural network [2], example-based models (e.g., k -nearest neighbor) [3], linear models [4], [5], support vector machine [6] and so on. Among these methods, a linear model called a centroid-based method is attractive since it has relatively less computation than other methods in both the learning and classification stages. Despite less computation time, a centroid-based method was shown in several literatures including those in [4], [5], to achieve relatively high accuracy. In this method, an individual class is modeled by weighting terms appearing in training documents assigned to the class. This makes classification performance strongly depend on term weighting applied in the model. Most previous works of a centroid-based method focused on weighting factors related to frequency patterns of terms or documents in the class. The most popular two factors are term frequency (tf) and inverse document frequency (idf). Some previous works, such as those in [7], [8], attempted to apply another factor called inverse class frequency (icf). However, the impact of this factor needs more investigation.

This paper investigates the usefulness of inverse class frequency in a more systematic way. The traditional term frequency and inverse document frequency utilize information within classes and in the whole collection of training data. Besides these two sources of information, information among classes so called inter-class information is expected to be useful. An inverse class frequency can utilize this source of information. The effectiveness of applying class frequency in term weighting and classification process is investigated. Three approaches are taken into account. The first one is to explore the effectiveness of inverse class frequency on the popular term weighting, i.e., $tf \times idf$, as a replacement of idf and an addition to $tf \times idf$. The second approach is to evaluate some functions, which are used to adjust the power of inverse class frequency. The other approach is to apply terms that are found in only one class or few classes, to improve classification performance, using two steps of classification. In the rest of this paper, section II presents centroid-based text categorization. Term weighting in centroid-based text classification by frequency-based patterns is given in section III. Section IV described three approaches for applying class frequency in text classification. The data sets and experimental settings are described in section V. In section VI, a number of experimental results are given. A conclusion is made in section VII.

II. Centroid-based Text Classification

In the centroid-based text categorization, a document (or a class) is represented by a vector using a vector space model with a bag of words (BOW) [9]. The simplest and popular one is applied term frequency (*tf*) and inverse document frequency (*idf*) in the form of $tf \times idf$ as a term weight for representing a document. In a vector space model, given a set of documents $D = \{d_1, d_2, ..., d_{|D|}\}$, a document d_j is represented by a document vector $\vec{d}_j = \{w_{1j}, w_{2j}, ..., w_{|T|j}\} = \{tf_{1j} \times idf_1, tf_{2j} \times idf_2, ..., tf_{|T|j} \times idf_{|T|}\}$, where w_{ij} is a weight assigned to a term t_i in a set of terms (T) of the document. In this definition, tf_{ij} is term frequency of a term t_i in a document d_j and idf_i is inverse document frequency, defined as $log(|D|/df_i)$. Here, $|D|$ is the total

DOI: 01.IJIT.02.02.57

number of documents in a collection and df_i is the number of documents, which contain the term t_i. Besides term weighting, normalization is another important factor to represent a document or a class. Class prototype $\vec{c}_k$ is obtained by summing up all document vectors in C_k and then normalizing the result by its size. The formal description of a class prototype $\vec{c}_k$ is $\sum_{dj \in C_k} \vec{d}_j / \| \sum_{dj \in C_k} \vec{d}_j \|$, where $C_k = \{ d_j | d_j$ is a document belonging to the class $c_k \}$. The simple term weighting is $\overline{tf} \times idf$ where $\overline{tf}$ is an average class term frequency of the term. The formal description of $\overline{tf}$ is $\sum_{dj \in C_k} tf_{ijk} / | C_k |$, where $| C_k |$ is the number of documents in a class c_k. Term weighting described above can also be applied to a query or a test document. In general, the term weighting for a query is $tf \times idf$. Once a class prototype vector and a query vector have been constructed, the similarity between these two vectors can be calculated. The most popular one is cosine distance [10]. This similarity can be calculated by the dot product between these two vectors. Therefore, the test document will be assigned to the class whose class prototype vector is the most similar to the vector of the test document.

III. Term Weighting in Centroid-based Text Classification by Frequency-based Patterns

This section presents concept of term weighting in centroid-based text classification using frequency-based patterns. Before the detail is described, some general characteristics of terms that are significant for representing a certain class, are listed below:

- A term tends to be a representative of a certain class, it should appear frequently with high occurrence frequency within the class.
- An important term seems to exist in relatively a few documents while a general term tends to appear in most documents in a collection.
- A crucial term seems to exist in only one class or few classes.

To realize these characteristics, frequency-based patterns in a training collection can be applied. Three main frequency factors are term frequency, document frequency and class frequency. The simplest and popular one is applied term frequency (tf) and inverse document frequency (idf) in the form of $tf \times idf$ for representing a document. In a centroid-based method, term frequency of a class vector comes from average value of term frequencies of documents in a class. Due to this, we use a symbol $\overline{tf}$ instead of tf. This represents an average class term frequency in a class. The $\overline{tf}$ is considered as an intra-class factor. A term that is important for a specific class, should have a high $\overline{tf}_{ik}$. The $\overline{tf}$ deals with the first property of the significant terms for classification. Term frequency alone may not be enough to represent the contribution of a term in a document. To achieve a better performance, the well-known inverse document frequency can be applied to eliminate the impact of frequent terms that exist in almost all documents. The idf_i is the inverse ratio of the number of training documents that contain the term t_i to the total number of training documents. It is usually applied in the form of the logarithm value of its original value. It deals with the second property of the significant terms. However, this tends to be true if those documents, which hold the terms, are in the same class. Inversely, if the distribution of the terms is uniform in all classes, idf becomes useless and is not helpful in classification. The idf is considered as a collection factor, i.e., its value is the same for a particular term, independent of arrangement of classes in a collection.

The first and second items can be coped by the conventional term frequency and inverse document frequency, respectively. However, for the third property, it is necessary to utilize other frequency factors. For this property, class frequency is expected to be useful. Class frequency enables the classifier to utilize the information among classes. It is considered as an inter-class factor. Some collections may have several possible organizations of documents (viewpoints). For example, a collection may be grouped into a set of classes based on its content (e.g., course, faculty, project, student, ...) or it may be grouped by its university (e.g., Cornell, Texas, Washington, Wisconsin,...). In these two cases, the collection factor (e.g., idf) of a term is identical while the inter-class factor of that term is varied. An important term of the specific category seems to exist in one class or only few classes (the third property of significant terms). Therefore, we can apply class frequency to represent important of that term. In the past, there were some works utilizing class frequency in general forms, such as logarithmic of inverse class frequency (icf). It is an analogous of idf. In some situations that we cannot calculate idf such as in [8], it is possible to set a sentence as a processing unit instead of a document, and hence icf replaces idf. Although the icf seems to be useful, it is not popular for applying into term weighting.

IV. Applying Class Frequency in Text Classification

In this paper, the usefulness of class frequency is

DOI: 01.IJIT.02.02.57

presented on both the concept and experiments. Three approaches are taken into account: (1) the effectiveness of icf in term weighting (2) several functions are applied to icf to enhance classification performance and (3) the algorithm to apply terms which are found in only one class or few classes, to improve classification performance, using two steps of classification. The detail is decribed as follow:

A. The Effectiveness of ICF in Term Weighting

The popular method to utilize class frequency is in the form of logarithmic function of inverse class frequency. The equation is shown below.

$$ICF_i = \log \frac{|C|}{cf_i} \quad (1)$$

Here cf_i is the number of classes that contain a term t_i. The effect of the icf is to promote a term which occurs in only few classes (later called *few-class terms*) and demote a term which appears in many classes (later called *most-class terms*). In the extreme case, where a term occurs only in one class $(cf_i = 1)$, the function obtains the maximum value. These terms are called *one-class terms*. We belief that few-class terms are quite important for classifying documents. Inverse class frequency promotes the importance of these terms. Inversely, when a term occurs in all classes $(cf_i = |C|)$, a function of inverse class frequency achieves a minimum value, i.e., 0. These terms are called *all-class terms*. They are considered as less important and may cause the classifier misclassify the document. However, their usefulness also depends on their term frequencies. The promotion of few-class terms is more effective than the demotion of most-class terms since it may demote the important terms that occur in all classes but useful for classifying documents. The icf can be included into term weighting. Two simple patterns are applied. The first pattern is $\overline{tf} \times icf$, i.e., substitution of the idf with icf. The other pattern is $\overline{tf} \times idf \times icf$, i.e., include the icf in the $\overline{tf} \times idf$. In normal situation, the number of documents is greater than the number of classes ($|D| > |C|$). This means the idf promotes the importance of terms more than the icf. When the logarithm of base 2 is used, the value of icf for a term is equal to 1 when the term is found in the half number of the total classes in a data set ($cf_i = |C|/2$). Therefore, the value of $icf > 1$ when $cf_i < |C|/2$ and $icf < 1$ when $cf_i > |C|/2$. If the value of icf more than 1, it promotes $\overline{tf}$ and $\overline{tf} \times idf$. On the contrary, it demotes $\overline{tf}$ and $\overline{tf} \times idf$ when its value less than 1.

B. Functions to modify ICF.

In this approach, several functions are applied to icf for adjusting level of promoting and demoting to $\overline{tf}$ and $\overline{tf} \times idf$. The first pattern is adjusting the power of icf. When the term weighting is $\overline{tf} \times idf \times icf$, a new term weighting is shown below

$$\overline{tf} \times idf \times icf^{\gamma} \quad (2)$$

Here, the γ is the power of icf. The icf promotes the $\overline{tf} \times idf$ when its value > 1. If the power of the icf is greater than 1, it is more powerful to promote or demote a term than a normal icf. On the other hand, when the power of the icf is less than 1, it is less powerful to promote or demote a term than a normal icf. The optimum value of the γ depends on data sets. In the first pattern, a term may be promoted or demoted. The second pattern considers only promotion of a term. An example of a term weighting formula is shown below

$$\overline{tf} \times idf \times (icf + 1) \quad (3)$$

From the equation, terms are found on all classes, are given a term weight of the $\overline{tf} \times idf$. The other terms are given with the values which are higher than the $\overline{tf} \times idf$.

C. Two-step Classification

In this approach of classification, class frequency of a term is applied to select representations of the class prototype and the test document vectors. The representative vectors are based on *n*-class terms. A representative vector of *n*-class terms, means it represents by terms which occur in *1,2,...,n* classes where $1 \le n \le |C|$. In the first step of classification, the *n* is set. A test document is classified to an appropriated class, based on *n*-class term. The rest of the test documents, which have not *n*-class terms as their representation, will be classified using all terms in the second step. For example, when *n*=1, i.e., the representation vectors of prototypes and test documents is only a set of terms that are found only in one class. A test document whose have one-class terms of a class, is automatically classified to that class in the first step. In the second step, the rest of the test documents will be classified by all terms.

V. Experimental Setting

To evaluate our concept about class frequency, three collections called WebKB, 7Sectors and 20Newsgroups are used. The WebKB, is a collection of web pages of computer

DOI: 01.IJIT.02.02.57

science departments in four universities with some additional pages from other universities. This collection can be viewed as two-dimensional viewpoints. In our experiment, we use the four most popular classes: *student, faculty, course* and *project*. This includes 4,199 web pages. Focusing on each class, five subclasses (the 2 *nd* dimension) are defined according to the university a web page belongs to: *Cornell, Texas, Washington, Wisconsin* and *miscellaneous*. We use this collection in two viewpoints: first dimension (WebKB1) and second dimension (WebKB2). The total number of data sets for this collection is two. Therefore, the effect of inverse class frequency with different numbers of classes on the same collection, can be evaluated. The second collection is the 7Sectors, a data set from CMU World Wide Knowledge Base Project from the CMU Text Learning Group. This data set has seven classes: *basic material, energy, financial, healthcare, technology, transportation* and *utilities*. The number of documents in this collection is 4,582. The third collection is 20Newsgroups (20News). The articles are grouped into 20 different UseNet discussion groups. It contains 19,997 documents and some groups are very similar.

All experiments were performed on closed test, .i.e., we used a training set as a test set. The performance was measured by classification accuracy defined as the ratio between the number of documents assigned with correct classes and the total number of test documents. As a preprocessing, some stop words (e.g., a, an, the) are excluded from all data sets. For the HTML-based data sets, all HTML tags (e.g., $< B >$, $< /HTML >$) were omitted from documents to eliminate the affect of these common words and typographic words. All headers are omitted from Newsgroups documents. A unigram model is applied in all experiments.

Three experiments are performed. The first experiment is to investigate effects of inverse class frequency on the four data sets of the three collections. The icf is multiplied to $\overline{tf}$ and $\overline{tf} \times idf$. The standard term weighting, $\overline{tf} \times idf$, is used as a baseline for comparison. The default term weighting for a test document is $tf \times idf$ on all experiments. In the second one, we investigate effects of the three functions on icf. In the last experiment, two-step classification is performed by using one-class terms and two-class terms in the first step.

VI. Experimental Results

A. Effect of ICF

In this first experiment, the effect of inverse class frequency are investigated on $\overline{tf}$ and $\overline{tf} \times idf$. Three different term weightings, i.e., $\overline{tf} \times idf$ (TFIDF), $\overline{tf} \times icf$ (TFICF) and $\overline{tf} \times idf \times icf$ (TFICFIDF), are applied to centroid-based classifiers. Four data sets are used for evaluation, i.e., WebKB1, WebKB2, 7Sectors and 20Newsgroups. Table I showed the result in forms of the classification accuracy.

Table I. Effect of ICF on the Four Data Sets

Classifier	Data Sets				
	WebKB1	*WebKB2*	*7Sectors*	*20News*	*Average*
TFIDF	86.54	94.83	76.45	79.09	84.23
TFICF	52.99	70.23	82,47	75.18	66.13
TFIDFICF	93.55	96.31	96.94	86.54	93.34

From the result, some observations can be made as follows. The performance of a prototype vector with $\overline{tf} \times idf$, is better than that of a prototype vector with $\overline{tf} \times icf$ on three data sets, except only the 7Sectors. The performance of classifiers with $\overline{tf} \times icf$ on the same collection but different viewpoints, i.e. WebKB1 and WebKB2, may be different with a large gap. The most effective term weighting for all data sets is $\overline{tf} \times idf \times icf$. It can be concluded that the icf express its usefulness when it is used to combine with $\overline{tf} \times idf$. In generally, the should not been used to replace the .

B. Funtions to Modify ICF

In this experiment, three functions are applied to , i.e., , and . These denote by (ICF+1), SqrtICF and (ICF)^2, respectively. The result is shown in Table II. Note that the result of TFIDFICF is represented again for comparison.
Some observations on several functions on ICF can be made as follows. The (ICF)^2 is the best by average on the four data sets. It can improve the performance of the classifier on three of four data sets, with the exception of WebKB2. On the 20News, the (ICF)^2 improve the performance with a gap of 4.43%. Although the performance of the classifier is the best by (ICF+1) on WebKB2, the average performance is less than the ICF. The average performance of the classifier by the SqrtICF is a little bit less than ICF.

C. Two-step Classification

In the last experiment, two-step classification is used. We apply *n*-class terms in the first step. The values of *n* are 1 and 2, i.e., one-class terms and two-class terms are investigated. For *n=1*, only a set of one-class terms is used as a representation of prototype and test document vectors. A test document that contains a set of one-class terms can be automatically classified to a class of those terms. In case of *n=2*, a set of one-class terms and two-class terms is used as a representation.

Table II. Effect of Different Functions applied to ICF

Classifier	Data Sets				
	WebKB1	*WebKB2*	*7Sectors*	*20News*	*Average*
TFIDFICF	93.55	96.31	96.94	86.54	93.34
TFIDF(ICF+1)	88.81	97.14	88.67	84.60	89.81
TFIDFSqrtICF	91.43	96.50	96.40	84.34	92.17
TFIDF(ICF)^2	94.64	95.55	97.32	90.97	94.62

DOI: 01.IJIT.02.02.57

The term weighting of the prototype vector is $\overline{tf} \times idf \times icf$. The rest of test documents are classified in the second step. Two term weightings are applied, i.e., $\overline{tf} \times idf$ and $\overline{tf} \times idf \times icf$. The result is shown in Table III.

The result shows a lot improvement of performance from classifiers when two-step classification is applied. In the first step, average classification accuracy on the four data sets is relatively high. Although performance in the first step of one-class terms representation is less than performance in two-class terms representation, the number of the rest documents of one-class terms representation is larger than that of two-class terms representation. The consequence of this is the number of documents that are assigned the correct class in the second step from one-class terms representation, is larger than that of two-class terms representation. When the classification process is completed, classification accuracy from all documents in a collection, beginning with one-class terms representation is higher than that of beginning with two-class terms representation. From the result, performance of classifiers with $\overline{tf} \times idf$ and $\overline{tf} \times idf \times icf$ in the second step, is quite competitive. Performance of the two-step classification is superior than that of the single step classification.

VII. Conclusion

This paper showed that class frequency was useful in centroid-based classification. Three approaches of a class frequency were investigated to exploit information among classes in a systematic way. The evaluation was conducted using various data sets. The first approach was to explore the effectiveness of inverse class frequency on the popular term weighting, i.e., $tf \times idf$, as a replacement of idf and an addition to $\overline{tf} \times idf$. The experimental results showed that classification accuracy of a classifier using the term weighting of $\overline{tf} \times idf \times icf$, outperformed those of $\overline{tf} \times idf$ and $\overline{tf} \times icf$. The second approach was to evaluate some functions, which were used to adjust the power of inverse class frequency.

Table III. Two-step Classification with One-class Terms and Two-class Terms

n	Method	Data Sets				
		WebKB1	*WebKB2*	*7Sectors*	*20News*	*Average*
1	Step 1: Auto. Assign	89.76	97.00	89.76	83.30	89.96
	Step 2: TFIDF	**97.83**	**99.67**	96.90	92.98	96.85
	Step 2: TFIDFICF	97.07	99.50	**97.80**	**93.62**	**97.00**
2	Step 1: TFIDFICF	91.88	96.76	94.10	88.65	92.85
	Step 2: TFIDF	93.81	96.95	96.94	91.65	94.84
	Step 2: TFIDFICF	93.64	96.93	97.01	91.64	94.81

From the result, the function of class frequency, which expressed the effect on both promoting and demoting some terms was more effective. Moreover, increasing the exponent of the icf, performance of classifiers is better on several data sets. The other approach was to apply terms, which were found in only one class or few classes, to improve classification performance, using two-step classification. The result showed obviously that, a lot improvement of performance from classifier over the single step classification. For conclusion, class frequency expressed its usefulness in text classification.

For the future works, effect of class frequency on other classification methods and performance of classifiers with class frequency on cross data sets, should be evaluated.

Acknowledgment

This work was funded by the Research and Development Institute, Silpakorn Univeristy via research grant SURDI 53/01/12.

References

[1] K. Nigam, A. K. McCallum, S. Thrun, and T. M. Mitchell, "Text classification from labeled and unlabeled documents using em," *Machine Learning*, vol. 39, no. 2/3, pp. 103–134, 2000.

[2] M. Ruiz and P. Srinivasan, "Hierarchical text classification using neural networks," *Information Retrieval*, vol. 5, no. 1, pp. 87–118, 2002.

[3] M. Kubat and M. Cooperson, Jr., "Voting nearest-neighbor subclassifiers," in *Proceedings of 17th International Conference on Machine Learning*, pp. 503–510, Morgan Kaufmann, San Francisco, CA, 2000.

[4] E.-H. Han and G. Karypis, "Centroid-based document classification: Analysis and experimental results," in *Proceedings of PKDD-00, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases*, (Lyon, FR), pp. 424–431, Springer-Verlag Publisher, 2000.

[5] V. Lertnattee and T. Theeramunkong, "Effect of term distributions on centroid-based text categorization," *Information Sciences*, vol. 158, pp. 89–115, 2004.

[6] T. Joachims, *Learning to Classify Text using Support Vector Machines*. Dordrecht, NL: Kluwer Academic Publishers, 2002.

[7] K. Cho and J. Kim, "Automatic text categorization on hierarchical category structure by using icf (inverse category frequency) weighting.," in *Proceedings of KISS-97, Conference of Korean Institute of Intelligent Systems*, pp. 507–510, 1997.

[8] Y. Ko and J. Seo, "Automatic text categorization by unsupervised learning," in *Proceedings of COLING-00, the 18th International Conference on Computational Linguistics*, pp. 453-459, Saarbrücken, DE, 2000.

[9] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, no. 5, pp. 513–523, 1988.

[10] A. Singhal, G. Salton, and C. Buckley, "Length normalization in degraded text collections," Tech. Rep. TR95-1507, 1995.

DOI: 01.IJIT.02.02.57